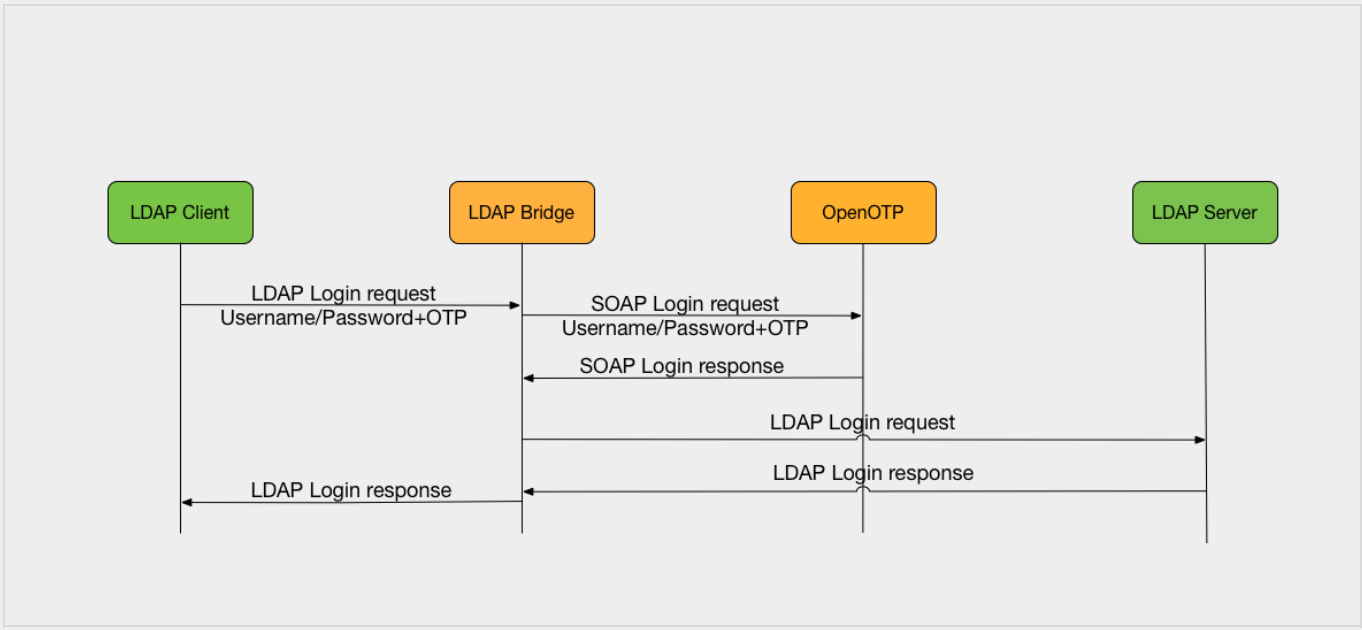
LDAP BRIDGE

LDAP Bridge

LDProxy LDAP



1. Product Overview

The main use-case of OpenOTP LDAP Bridge is enabling enterprise applications that use LDAP as an external authentication mechanism to work with OpenOTP. LDAP Bridge allows authentication to be delegated to an OpenOTP server transparently, without changing the LDAP back-end. From the client applications perspective, the main change is that it will use the LDAP Bridge as an LDAP server, instead of the backend-end LDAP server.

LDAP Bridge works by relaying LDAP messages to a back-end LDAP server. It intercepts user bind (LDAP authentication) operations and makes an OpenOTP call to authenticate the request with OpenOTP. It then sets the result of the bind request to the authentication result of the OpenOTP call.

One drawback of LDAP protocol is that LDAP bind does not support challenge-response or interactive user dialogue, which means that all authentication factors must be passed concatenated in one unique login request. Like RCDevs' OpenOTP RADIUS Bridge, LDAP Bridge is not designed to be exposed to the internet, but rather to sit beside WebADM, or in a DMZ.

2. System Requirements

LDAP Bridge runs on Linux 64bit operating systems with GLIBC ≥ 2.5. The installation package contains all the required dependencies allowing LDAP Bridge to run on any Linux system without any other requirement.

LDAP Bridge requires a working OpenOTP+WebADM installation (version ≥ 1.4) connected to an LDAP backend.

The LDAP Bridge can be run on the same server as OpenOTP and WebADM. A standalone LDAP Bridge should meet the following requirements:

- Running a Linux distribution with Glibc ≥ 2.5 installed (RedHat, CentOS, SUSE, Debian, Ubuntu).
- At least a 1 GHz x86-64 processor (two cores or vCPUs recommended).

- 512 MB of RAM.
- At the very least 20MB of free disk space.

3. Installation

3.1 Install with Yum Repository

On a RedHat, CentOS or Fedora system, you can use our repository, which simplifies installation and updates.

Add the repository:

```
yum install https://www.rcdevs.com/repos/redhat/rcdevs_release-1.0.0-0.noarch.rpm
```

Clean the yum cache and install LDAP Bridge:

```
yum clean all
yum install ldproxy
```

3.2 Install with Debian Repository

On a Debian system, you can use our repository, which simplifies installation and updates.

Add the repository:

```
wget https://www.rcdevs.com/repos/debian/rcdevs-release_1.0.0-0_all.deb
apt-get install ./rcdevs-release_1.0.0-0_all.deb
```

Clean the cache and install WebADM with all WebApps & Services:

```
apt-get update
apt-get install ldproxy
```

3.3 Install using the Self-Installer

You first need to download and install the LDAP Bridge software package. You can download *OpenOTP LDAP Bridge* on the RCDevs Website and copy it to your server. You can copy the package file to the server with WinSCP or scp. Then connect via SSH to your server, uncompress and run the self-installer package with:

```
gunzip ldproxy-1.2.*.sh.gz
bash ldproxy-1.2.*.sh
```

The installation process will automatically run the console-based setup script in `bin/setup`.

4. Configuration

Once the package is installed, you can run the setup script:

```
[root@ldproxy ~]# /opt/ldproxy/bin/setup
Checking system architecture...Ok
```

You insert the hostname of the LDAP Bridge server for the certificate generation:

```
Enter the server fully qualified hostname (FQDN): ldproxy.test.local
Enter LDAP server IP or hostname [localhost]: backend_ldap.test.local
Enter LDAP server port [389]:
389
Enter LDAP protocol (ldap/ldaps) [ldap]:
ldap
```

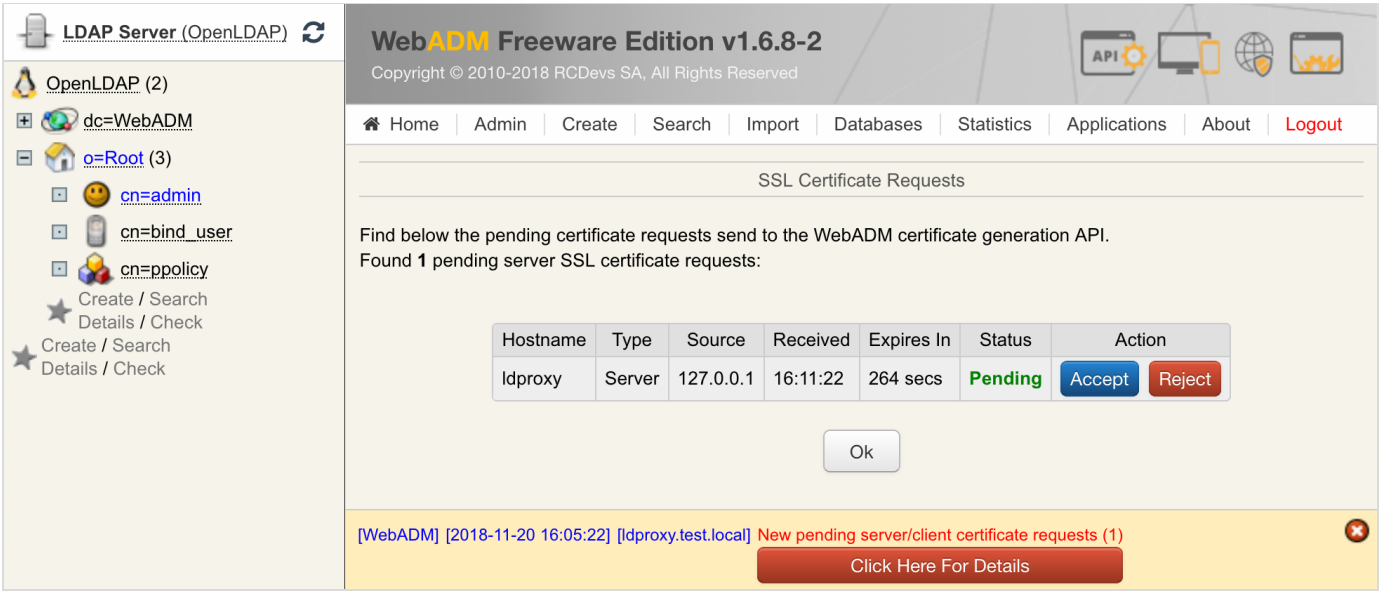
If the login mode defined in *openotp* is *OTP*, then you need to configure a bind account in ldproxy, if it's only *LDAPOTP*, you can keep it empty because *ldproxy* is able to forward the LDAP request with the correct password to the LDAP backend:

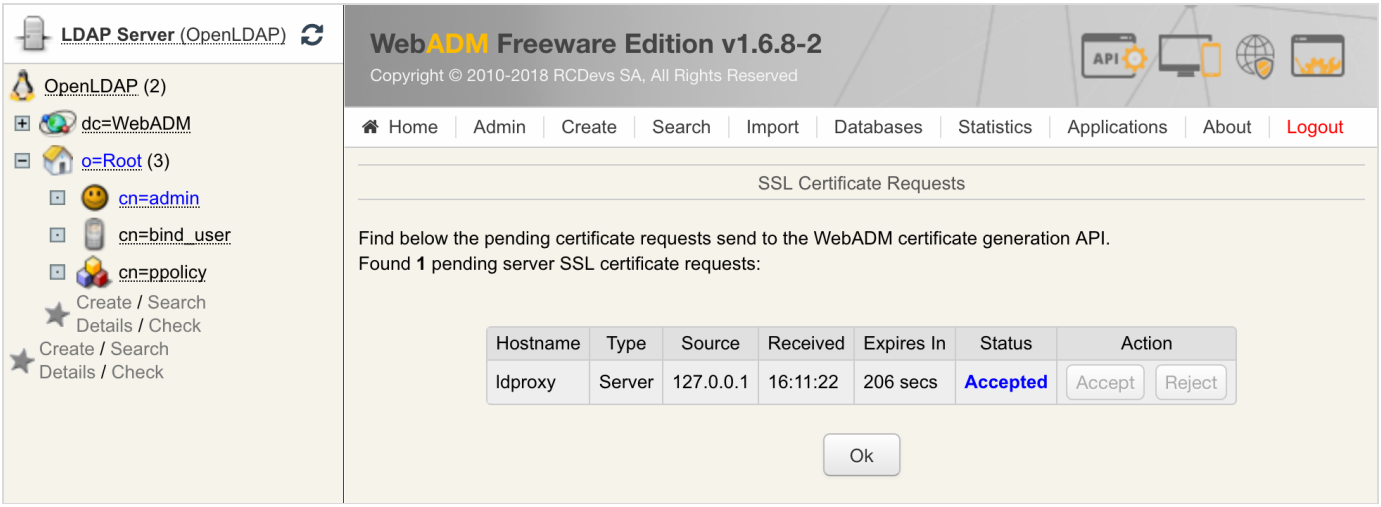
```
Enter a bindable LDAP account from the back-end with no specific permission:
cn=bind_user,cn=users,dc=test,dc=local
Enter the LDAP account password:
```

You enter the IP of the WebADM server:

```
Enter WebADM server IP or hostname [localhost]: webadm1.test.local
Found two server URLs:
> URL1: https://webadm1.test.local:8443/openotp/
> URL2: https://webadm2.test.local:8443/openotp/
Retrieving WebADM CA certificate... Ok
The setup needs now to request a signed SSL server certificate.
This request should show up as pending in your WebADM interface and an administrator
must accept it!
Waiting 5 minutes for approbation... Ok
```

You connect to the WebADM interface and approve the certificate request:






```
Updating OpenOTP configuration file... Ok
Setting file permissions... Ok.
Starting OpenOTP LDAP Bridge... Ok
Do you want OpenOTP LDAP Bridge to be automatically started at boot (y/n)[y]?
y
Adding systemd service... Ok
Do you want to register OpenOTP LDAP Bridge logrotate script (y/n)[y]?
y
Adding logrotate script... Ok
OpenOTP LDAP Bridge has successfully been setup.
```

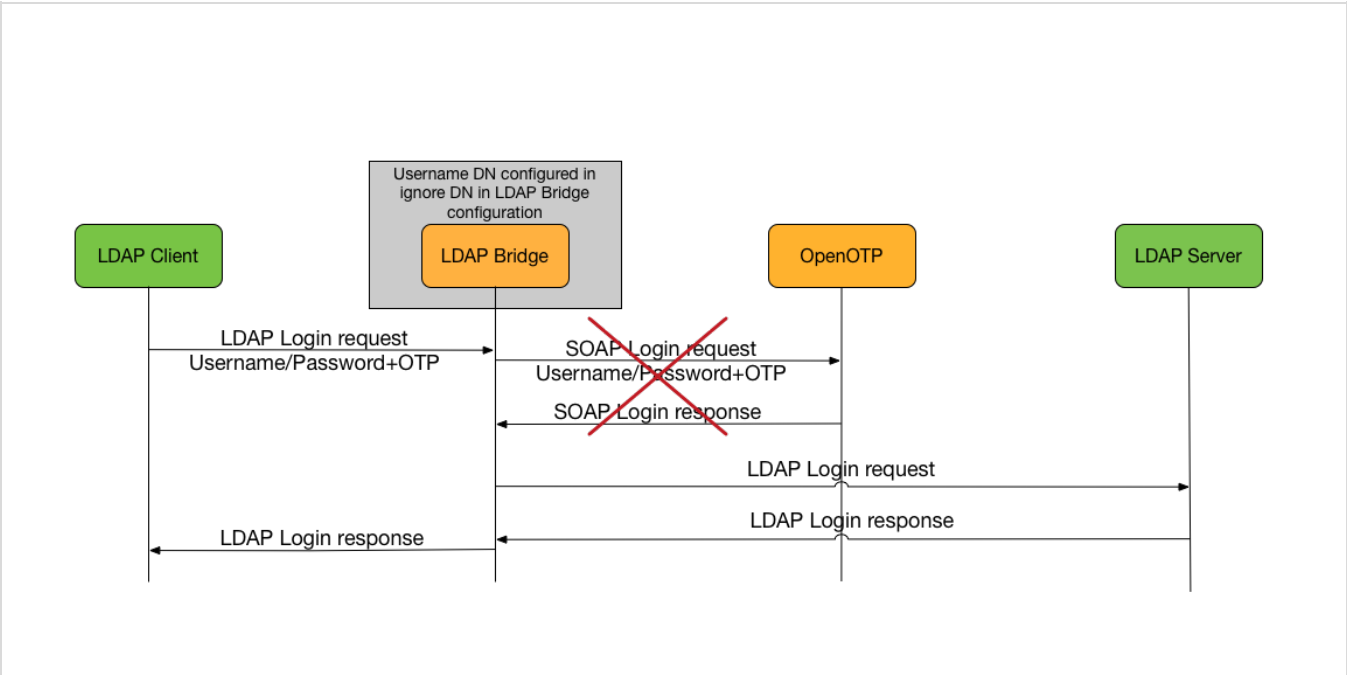
You can use `ldapsearch` for testing. If it's not already available, you can install it with

`yum install openldap-clients` on CentOS. In this example, the user *john* is reading informations about himself, his password is *password* and his OTP is *637991*:

```
[root@ldproxy ~]# ldapsearch -H ldap://ldproxy.test.local:10389 -D  cn=john,o=Root -w
password637991 -b cn=john,o=Root dn
# extended LDIF
#
# LDAPv3
# base <cn=john,o=Root> with scope subtree
# filter: (objectclass=*)
# requesting: dn
#

# john, Root
dn: cn=john,o=Root

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

You can also define settings manually in `ldproxy.conf`:

- *cert_file*: OpenOTP client certificate.
- *cert_password*: OpenOTP client certificate's password.
- *client_id*: the client ID that will be set in every request to OpenOTP, who can then match requests to a client policy with the same name (or alias).
- *denied_dn*: a list of users who are not allowed to be authenticated by OpenOTP, they will receive an authentication failure.
- *domain*: the WebADM domain that will be set on every request to OpenOTP.
- *ignored_dn*: a list of users who don't need to use OpenOTP, the authentication is not redirected to the OpenOTP server.



- *server_policy*: the load-balancing policy of requests between OpenOTP servers, if two servers are defined in *server_url*.
- *server_url*, *server_url1*, *server_url2*: openotp server url.
- *soap_timeout*: the time in seconds without before LDproxy's connection to OpenOTP times out. The LDAP clients of LDproxy must have a higher tolerance than soap_timeout to timeouts when connecting to LDproxy.
- *status_cache*: the time in seconds between health polls of the backend OpenOTP servers.
- *user_settings*: which are the public OpenOTP settings that will be passed in every request. OpenOTP must be configured with "Allow Request Settings" in WebADM. These settings will have priority over any settings defined on the users, groups, client policies and OpenOTP configuration.

Upgrades of LDAP Bridge will overwrite the file `/opt/ldproxy/conf/ldproxy.conf.default`, which will indicate the default values for any new configuration directive added by the upgrade. If new directives or any significant change has been added, it will be mentioned in `/opt/ldproxy/RELEASE_NOTES`.

4.1 LDAP sections

These sections contain per LDAP backend configurations:

- *bind_dn* and *bind_pw*: Proxy user used for OTP user binds to the LDAP back-end. It must be a bindable LDAP account with no specific permission.
- *domain*: the WebADM domain that will be set on every request to OpenOTP.
- *name*: the name shown in logs.
- *security*: Specify a set of security strength factors
- *suffix*: base dn corresponding to the backend, it should not be included in a previous defined ldap backend.
- *tls*: tls configuration `{[try-]start|[try-]propagate|ldaps}` `[tls_cert=<file>]` `[tls_key=<file>]` `[tls_cacert=<file>]` `[tls_cacertdir=<path>]`

`[tls_reqcert=never|allow|try|demand]` `[tls_ciphersuite=<ciphers>]` `[tls_crlcheck=none|peer|all]`.

- *uri*: ldap backend uri.

4.2 Client sections

`ldproxy.conf` can also contain client sections, which can, for requests coming from a specific IP or a subnet, override *client_id*, *domain* and *ignored_dn*.

4.3 LDAP Ports

LDAP Bridge provides the LDAP service over the following ports:

- TCP 10389 for un-encrypted LDAP and TLS.
- TCP 10636 for LDAP over SSL.

The LDAP Bridge's default listening network interface and ports can be changed by creating an environment file `/opt/ldproxy/conf/ldproxy.env` with the following configurations:

```
##This is ldproxy.env example

INTERFACE=0.0.0.0
PORT_STD=10389
PORT_SSL=10636
```

5. Maintenance and Troubleshooting

This section should cover your common administrative tasks concerning LDAP Bridge. For additional support, you can contact RCDevs' commercial support if you are a client or our Google Group if you are using the freeware edition of OpenOTP.

5.1 Starting and Stopping

If during the setup, you've let the installer set the LDProxy init scripts and systems service files on your machine, the LDAP Bridge should start at machine boot. You should also be able to start and stop the LDAP Bridge through your distribution's usual commands, such as `systemctl start ldproxy` for distributions using systemd like RedHat Enterprise Linux 7.

Alternatively, you can use

```
/opt/ldproxy/bin/ldproxy start | restart | stop | debug
```

5.2 Backup and Restore

You can backup and restore easily the configuration with these commands:

```
/opt/ldproxy/bin/backup ldproxy.bkp.gz
```

```
/opt/ldproxy/bin/restore ldproxy.bkp.gz
```

5.3 Upgrading and Un-Installing

If LDAP Bridge was installed using RCDevs repository, it will be updated with the system when you will execute `yum update` or `apt-get upgrade`.

If it was installed with the tar file, you can download and install it as you did for your first installation. The installer will offer you the option of upgrading your installation.

Be aware that, to do so, the installer will stop LDProxy. As a matter of principle, you should back up the `/opt/ldproxy/` directory before the upgrade. You can then restore the directory if anything breaks and restart the LDProxy service.

The installer also gives you the option of removing an existing LDProxy installation.

You can reset your installation by executing `/opt/ldproxy/setup reset`, which removes any init, systemd and logrotate files the installer put on the machine. This will also remove the log files, SSL certificate and secret key.

5.3.1 Upgrading from 1.1.x to 1.2.x

The version 1.2.0 includes the support of multiple LDAP backends. You need to change some settings manually.

Before the update, keep a backup of `/opt/ldproxy/conf` folder, then run the update.

Once it is done, rename `default_domain` parameter as `domain`. Create also an LDAP section at the end of `ldproxy.conf` and move the config for the LDAP backend in it:

```
ldap {
 uri     "ldaps://ldap.backend1:636,ldaps://ldap.backend2:636"
 suffix  ""
 bind_dn "cn=test,dc=webadm"
 bind_pw "Password123"
 domain  "Acme"
}
```

You can use `ldproxy.conf.default` as an example of the new configuration.

5.3.2 Upgrading from 1.0.x to 1.1.x

The version 1.1.0 includes several changes. You probably need to change some settings manually.

Before the update, keep a backup of `/opt/ldproxy/conf` folder, then run the update.

Once it is done, if you don't have changed the default port, you probably need to change it. The previous version used 389 and 636 and the new version 10389 and 10636. If you want to continue to use 389 and 636, you need to create `/opt/ldproxy/conf/ldproxy.env` with the following content:

```
PORT_STD=389
PORT_SSL=636
```

You need also to copy `uri` and `ignored_dn` from `slapd.conf` to `ldproxy.conf`. `uri` is now called `ldap_uri1`.

To finish, you need to replace `denied_usernames` with `denied_dn`, replace `nolock_usernames` with `nolock_dn`, replace all usernames with their distinguished name and remove `uid_attribute` in `ldproxy.conf`.

You can use `ldproxy.conf.default` as an example of the new configuration.

5.4 Troubles and Known Issues

You can start LDAP Bridge in debug mode to get a verbose output of what the proxy does on your terminal.

```
/opt/ldproxy/bin/ldproxy debug <loglevel>
```

If you omit `<loglevel>`, it will be set to `stats` by default, but you can also choose another log level:

log level	Description
any	enable all debugging
trace	trace function calls
packets	debug packet handling
args	heavy trace debugging
conns	connection management
BER	print out packets sent and received
filter	search filter processing
config	configuration processing
ACL	access control list processing
stats	stats log connections/operations/results
stats2	stats log entries sent
shell	print communication with shell backends
parse	print entry parsing debugging
sync	syncrepl consumer processing
none	only messages that get logged whatever log level is set